REMARKS

Claims 1 and 9 are pending in the application. Claims 1-17 have been rejected.

Claims 1-8 and 11-15 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Metzger et al., U.S. Patent No. 7,269,827 (Metzger).

Claim 1 is allowable over Metzger

The present invention, as set forth by independent claim 1 relates to a method of determining an execution order for machine instructions to reduce spill code, the method comprising the steps of scheduling the machine instruction for which an amount by which a size of a committed set of machine instructions would increase upon the scheduling of the machine instruction is smallest from machine instructions that are ready for scheduling, and determining an execution order for the machine instructions to reduce spill code.

The present invention, as set forth by amended independent claim 1, relates to a method of determining an execution order for machine instructions to reduce spill code. The method includes scheduling the machine instruction from machine instructions that are ready for scheduling for which an amount by which a size of a committed set of machine instructions would increase upon the scheduling of the machine instruction is smallest, the committed set of machine instructions including any machine instruction that is already scheduled and any machine instruction that is a descendent from an already scheduled machine instruction, for each of the machine instructions ready for scheduling, the amount being determined by identifying descendent machine instructions of each of the machine instructions, determining which of the descendent machine instructions and the machine instructions is not in the committed set of machine instructions, determining an execution order for the machine instructions to reduce spill code, a given machine instruction being considered ready for scheduling when scheduling of the given machine instruction as a next machine instruction would not cause an erroneous programmatic result, and undertaking the method when a risk of register overcommittedness exceeds a certain threshold where the threshold is exceeded when processor register availability drops below a particular threshold.

Metzger relates to compiling high level code. Metzger discloses integrating allocation of registers, scheduling instructions, and selecting code functions to produce an intermediate representation of a high level code segment with scheduled instructions. Additionally, Metzger discloses a modular conflict handler to resolve register and scheduler conflicts as may be required in compiling the high level code. Additionally, Metzger discloses a modular transformation interface to invoke analyzers to generate a compiled version of the high level code.

Metzger, taken alone or in combination, does not teach or suggest a method of determining an execution order for machine instructions to reduce spill code where the method includes, scheduling the machine instruction from machine instructions that are ready for scheduling for which an amount by which a size of a committed set of machine instructions would increase upon the scheduling of the machine instruction is smallest, the committed set of machine instructions includes any machine instruction that is already scheduled and any machine instruction that is a descendent from an already scheduled machine instruction, for each of the machine instructions ready for scheduling, the amount being determined by identifying descendent machine instructions of each of the machine instructions, determining which of the descendent machine instructions and the machine instructions is not in the committed set of machine instructions, determining an execution order for the machine instructions to reduce spill code, a given machine instruction being considered ready for scheduling when scheduling of the given machine instruction as a next machine instruction would not cause an erroneous programmatic result, and undertaking the method when a risk of register overcommittedness exceeds a certain threshold, the threshold is exceeded when processor register availability drops below a particular threshold, all as required by Claim 1. Accordingly, Claim 1 is allowable over Metzger.

Claims 9 is allowable over Widell

Claims 9-10 and 16-17 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Widell et al., U.S. Publication No. 2005/0055490 (Widell)

The present invention, as set forth by amended independent claim 9, relates to a method of determining an execution order for machine instructions to reduce spill code. The method includes the steps of in a first bit vector containing one bit to represent each machine instruction to be scheduled, setting those bits for which the represented machine instruction is not committed, resetting the remaining bits, for the each machine instruction to be scheduled that is ready for scheduling where in a second bit vector also having one bit to represent each machine instruction to be ordered in the same sequence as in the first bit vector, setting those bits for which the represented machine instruction is a descendant of the each machine instruction that is ready for scheduling, and resetting the remaining bits, performing a bitwise AND operation of the first bit vector and the second bit vector to create a third bit vector; prior to performing the bitwise AND operation, setting in the second bit vector the bit for which the represented machine instruction is the each machine instruction that is ready for scheduling, determining the number of set bits in the third bit vector, and selecting for execution the machine instruction for which the third bit vector contains a minimum number of set bits.

Widell relates to mechanisms for handling and detecting collisions between threads that execute computer program instructions out of program order. According to an embodiment of the present invention each of a plurality of threads are associated with a respective data structure comprising a number of bits that correspond to memory elements (m.sub.0, m.sub.1, m.sub.2, m.sub.n) of a shared memory. When a thread accesses a memory element in the shared memory, it sets a bit in its associated data structure, which bit corresponds to the accessed memory element. This indicates that the memory element has been accessed by the thread. Collision detection may be carried out after the thread has finished executing by means of comparing the data structure of the thread with the data structures of other threads on which the thread may depend.

While Widell discloses bit vectors, Widell does not disclose or suggest a bit vector containing one bit to represent each machine instruction to be scheduled, setting those bits for which the represented machine instruction is not committed and resetting the remaining bits, as required by claim 9. Additionally, Widell does not disclose or suggest a second bit vector also having one bit to represent each machine instruction to be ordered in the same sequence as in the first bit vector, setting those bits for which the represented machine instruction is descendant of

each machine instruction that is ready for scheduling and resetting the remaining bits, as required by claim 9. Accordingly, Widell could not disclose or suggest performing a bitwise AND operation on the first bit vector and the second bit vector to create a third bit vector or determining a number of set bits in the third bit vector, much less selecting for execution the machine instruction for which the third bit vector contains a minimum number of set bits, as required by claim 9. Additionally, Widell does not disclose or suggest prior to performing a bitwise AND operation, setting in a second bit vector a bit for which the represented machine instruction is each machine instruction that is ready for scheduling, as is required by claim 9. Thus, for each of these reasons, claim 9 is allowable over Widell.

CONCLUSION

In view of the amendments and remarks set forth herein, the application is believed to be in condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the examiner is requested to telephone the undersigned.

The Commissioner is hereby authorized to charge Deposit Account No. 090461 for any fees due and credit any overpayments to same.

I hereby certify that this correspondence is being electronically submitted to the COMMISSIONER FOR PATENTS via EFS on April 7, 2008.

/Stephen A. Terrile/

Attorney for Applicant(s)

Respectfully submitted,

/Stephen A. Terrile/

Stephen A. Terrile
Attorney for Applicant(s)
Reg. No. 32,946